Northumbria Research Link

Advancing Ensemble Learning Performance through Data Transformation and Classifiers Fusion in Granular Computing Context

Han Liu[a,*], Li Zhang[b]

[a]*School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Cardiff, CF24 3AA, United Kingdom*
[b]*Department of Computer and Information Sciences, Faculty of Engineering and Environment, Northumbria University, Newcastle NE1 8ST, United Kingdom*

Abstract

Classification is a special type of machine learning tasks, which is essentially achieved by training a classifier that can be used to classify new instances. In order to train a high performance classifier, it is crucial to extract representative features from raw data, such as text and images. In reality, instances could be highly diverse even if they belong to the same class, which indicates different instances of the same class could represent very different characteristics. For example, in a facial expression recognition task, some instances may be better described by Histogram of Oriented Gradients features, while others may be better presented by Local Binary Patterns features. From this point of view, it is necessary to adopt ensemble learning to train different classifiers on different feature sets and to fuse these classifiers towards more accurate classification of each instance. On the other hand, different algorithms are likely to show different suitability for training classifiers on different feature sets. It shows again the necessity to adopt ensemble learning towards advances in the classification performance. Furthermore, a multi-class classification task would become increasingly more complex when the number of classes is increased, i.e. it would lead to the increased difficulty in terms of discriminating different classes. In this paper, we propose an ensemble learning framework that involves transforming a multi-class classi-

*Corresponding author
Email addresses: LiuH48@cardiff.ac.uk (Han Liu), li.zhang@northumbria.ac.uk (Li Zhang)

fication task into a number of binary classification tasks and fusion of classifiers trained on different feature sets by using different learning algorithms. We report experimental studies on a UCI data set on Sonar and the CK+ data set on facial expression recognition. The results show that our proposed ensemble learning approach leads to considerable advances in classification performance, in comparison with popular learning approaches including decision tree ensembles and deep neural networks. In practice, the proposed approach can be used effectively to build an ensemble of ensembles acting as a group of expert systems, which show the capability to achieve more stable performance of pattern recognition, in comparison with building a single classifier that acts as a single expert system.

Keywords: Machine Learning, Ensemble Learning, Classification, Bagging, Boosting, Random Forests

---

1. Introduction

Machine learning is a branch of artificial intelligence, which can be typically categorized into supervised learning and unsupervised learning. Supervised learning is generally aimed at learning from labelled data, which means that each training instance is labelled by domain experts. In contrast, unsupervised learning is generally aimed at learning from unlabelled data, which means that none of the training instances is provided with a label. In practice, supervised learning is involved in classification and regression tasks, and unsupervised learning is involved in association and clustering tasks. The rest of this paper will focus on classification tasks.

In the context of machine learning, classification can be achieved by training classifiers that can be used to classify new instances. In order to train high quality classifiers, it is crucial to ensure good features to be extracted from original data. In popular application areas, such as text classification and image processing, there are various feature extraction methods available leading to different types of features. However, instances could be highly diverse even if they belong to the same class, i.e. some instances may present one type of features but other instances present another type of features. From this point of view, it is difficult to decide which method of feature extraction should be adopted towards transforming a set of raw data into a good feature set, since it is very likely that a classifier trained on one feature set is capable of classifying some but not all instances. In order to increase

the chance of correctly classifying each single instance, it is necessary to make sure that each instance can be classified by using a classifier that is learned from features relevant to the instance. On the basis of the above argumentation, it is necessary to undertake the classification task in the context of ensemble learning, i.e. several feature sets are extracted from the original data and a classifier is trained on each feature set to be fused with other classifiers trained on other feature sets.

On the other hand, it is usually the case in real applications that each learning algorithm has its own advantages and disadvantages, so different algorithms usually show different suitability for training classifiers on different feature sets. In other words, it is fairly difficult to have a general conclusion that a learning algorithm is capable of training good classifiers on all feature sets. Again, due to the diversity among instances, it is highly possible that different classifiers trained on the same feature set by using different algorithms show inconsistent classification confidence on different instances, although one classifier would have the highest overall confidence on all the instances. For example, there are two classifiers A and B and it is very normal that classifier A has a higher confidence on instance 1 but classifier B has a higher confidence on instance 2. From this point of view, it is necessary to adopt fusion of multiple classifiers trained on the same feature set by using different learning algorithms.

Furthermore, in the case of multi-class classification, the increase of the number of classes would usually lead to increasing difficulty in discriminating between classes. For example, in a binary classification task, it is much easier to see the tendency (degree of discrimination) that a classifier is biased towards one class and against the other one through looking at the probability distribution between the two classes, but the discrimination is obviously more difficult when there are multiple classes. Also, there is usually uncertainty in reality on how well a training set can represent a full population of a problem domain. Therefore, the uncertainty is likely to result in incorrect estimation of the probability distribution among classes. From this point of view, it is necessary to transform a mutli-class classification task into n binary classification tasks, where n is the number of classes.

In this paper, we propose an ensemble learning framework in the setting of granular computing to address the three points mentioned above. In other words, the proposed framework involves transforming the task of training a multi-class classifier into several separate tasks of training n binary classifiers respectively for the n given classes. For training each binary classifier,

the framework is designed to involve primary fusion of classifiers trained on different feature sets by using the same learning algorithm and secondary fusion of the previously fused classifiers resulting from the primary fusion stage. The contributions of this paper include the following:

- We propose a systematic framework of ensemble learning towards in-depth training and fusion of classifiers, which is naturally inspired from the system theory that can be used for creating a group of expert systems with collaborations to each other.
- We demonstrate a novel application of granular computing concepts in the context of ensemble learning, i.e. the proposed ensemble learning framework shows characteristics of granular computing based data processing, which can benefit the development of intelligent system of data processing.
- We compare the proposed ensemble learning framework with popular ensemble and deep learning approaches as well as each standard learning algorithm that is used to train base classifiers as part of an ensemble, in terms of classification performance, and the experimental results show the proposed ensemble learning framework effectively leads to considerable advances in the classification performance.

The rest of this paper is organized as follows: Section 2 provides a review of existing approaches of ensemble learning and an overview of granular computing concepts and techniques. In Section 3, we illustrate the proposed framework of ensemble learning and justify its significance in advancing the classification performance. In Section 4, we report an experimental study conducted by using a UCI data set and a facial expression data set and the results are discussed in depth to show the effectiveness of the proposed approach of ensemble learning. In Section 5, the contributions of this paper are summarized, and further directions are suggested towards advancing this research area in the future.

2. Related Work

In this section, we provide a review of ensemble learning approaches and identify the limitations of these approaches. Also, we provide an overview of granular computing concepts and techniques.

2.1. Review of Ensemble Learning Approaches

Ensemble learning is aimed at training an ensemble of classifiers that can be combined for advancing the overall classification performance, in comparison with use of a single classifier. In particular, an ensemble learning task involves ensembles creation and classifiers fusion.

In order to create an ensemble of higher performance, it is necessary to ensure the two points (Zhou, 2012): a) each single classifier in the ensemble must not be bad; b) the classifiers in the ensemble need to be highly diverse, i.e. different classifiers should result in different sets of incorrectly classified instances and the ideal outcome is that for each instance at least one classifier gives correct classification. Ensembles creation can be achieved by training base classifiers in parallel or sequential training of these classifiers. Two popular approaches of ensembles creation are referred to as Bagging and Boosting, respectively.

Bagging, which was developed in Breiman (1996), follows the parallel ensemble learning approach. In particular, the Bagging approach involves random sampling of training data with replacement, which indicates that some instances may be selected more than once but other instances may never be selected. On average, each sample is expected to contain 63.2% of the training instances (Tan et al., 2005; Liu and Gegov, 2015). The random sampling would result in n training samples and each base classifier is trained on one of the n samples to become a member of the ensemble. In the testing stage, each of the base classifiers in the ensemble makes an individual classification first for each instance and then the outputs of these base classifiers are fused to make the final classification for the instance through majority voting. Due to the case that different samples cover different parts of the training set, it is likely that the base classifiers trained on these samples are diverse leading to advances in the classification. The whole procedure of the Bagging approach is illustrated in Fig. 1.

A popular example of the Bagging approach is referred to as Random Forests (RF), which is aimed at creation of decision tree (DT) ensembles (Breiman, 2001). In practice, it has been proven experimentally that the RF method can effectively lead to advances in classification performance, in comparison with the standard DT learning method (Kononenko and Kukar, 2007; Tan et al., 2005). The advances are mainly due to the adoption of random data sampling and the random subspace method (Ho, 1998), leading to the increase of the diversity among the trained DT classifiers (Zhou, 2012; Melville and Mooney, 2005).

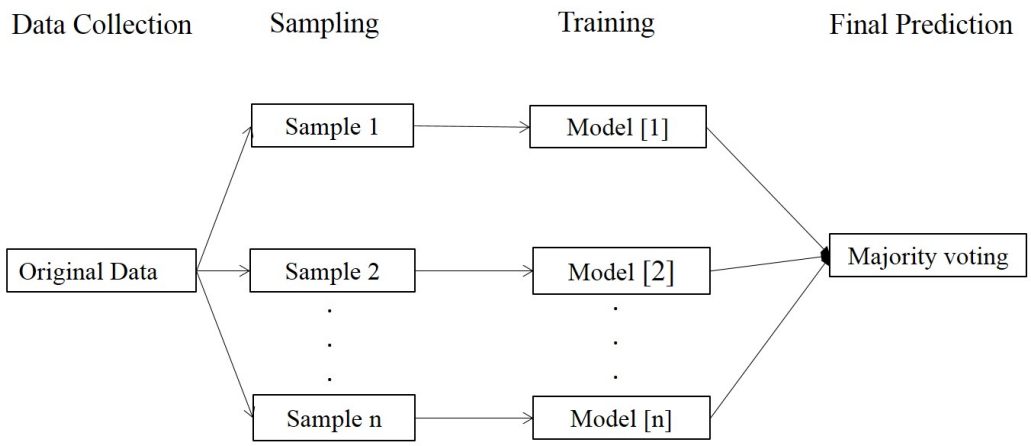

Figure 1: The Procedure of Bagging (Liu et al., 2016)

Boosting, which was developed in Freund and Schapire (1996), follows the sequential ensemble learning approach. In particular, the Boosting approach involves n iterations of training, i.e. a base classifier is trained at each iteration. The training of each base classifier h_t at iteration t depends on the experience gained from the its former classifier trained at iteration $t-1$ (Li and Wong, 2004). In other words, the latter classifier is trained by focusing on learning from the instances incorrectly classified by its former classifier. In this context, each base classifier is assigned a weight depending on its accuracy estimated by using validation data. The stopping criteria are satisfied while the error rate is equal to 0 or greater than 0.5 (Li and Wong, 2004). In the testing stage, each of the n base classifiers makes an independent classification in a similar way to Bagging, but the final classification is made by fusing the outputs of the base classifiers through weighted voting instead of majority voting. Due to the case that different base classifiers are trained by focusing the corresponding learning tasks on different parts of the training set, it is likely that the base classifiers trained at different iterations are diverse, leading to advances in the classification performance. A popular example of Boosting is referred to as Adaboost, which is illustrated below (Freund and Schapire, 1996):

Given: (x_1, y_1), ..., (x_m, y_m) where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, ..., T$:

- Train weak learner using distribution D_t.

- Get weak hypothesis $h_t : X \rightarrow \{\text{-1, +1}\}$ with error $\epsilon_t = Pr_i[h_t(x_i) \neq y_i]$.

- Choose $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$.

- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \left\{ \begin{array}{ll} e^{-\alpha_t}, & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t}, & \text{if } h_t(x_i) \neq y_i \end{array} \right\}$$

$$= \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

  where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis: $H(x) = sign(\sum^T \alpha_t h_t(x))$

In the above illustration, x_i indicates an input vector and y_i indicates the class label assigned to x_i, where i is the index of an instance. Also, X and Y represent the domain and range of the given data set respectively. In addition, the distribution D_t reflects how each instance is weighted at each particular iteration of the procedure for the Adaboost method. The symbol t represents the number of the current iteration and α_t represents the weight of the classifier learned at the iteration t.

Another example of Boosting is referred to as Gradient Boosted Trees (GBT), which has been popularly used for creation of DT ensembles for advancing the performance of decision tree learning (Ogutu et al., 2011).

In terms of classifiers fusion in the testing stage, there are several rules of fusion as studied in Kittler et al. (1998); Kuncheva (2002); Kittler and Alkoot (2003); Tax et al. (2000, 1997), which can impact the performance of classification. As proposed in Xu et al. (1992), the outputs of base classifiers can be measured in three levels:

- Abstract: the class label of each classifier is used as an output
- Rank: a ranking list of class labels is provided as the output of a classifier
- Measurement: the posterior probability of each class is provided as an output of a classifier

The rank level outputs are generally used for multi-class classification tasks. The outputs in the abstract or measurement level are commonly used for classifiers combination in both binary and multi-class classification tasks. In the abstract level, the rule of combination is referred to as vote, which simply counts the votes for each class and outputs the class that obtains the most votes. In the measurement level, some common rules of combination include sum, mean, max, min, median and product.

Given a n-class classification problem: $y \in \{c_1, c_2, ..., c_n\}$, and m classifiers $\{h_1, h_2, ..., h_m\}$ are trained in a feature space $D : \{x_1, x_2, ..., x_k\}$, the combination rules are defined in Eqs. (1)-(6).

$$P_{Ensemble}(c_i|x_1, x_2, ...x_k) = \sum_{j=1}^{m} P_{h_j}(c_i|x_1, x_2, ...x_k) \tag{1}$$

$$P_{Ensemble}(c_i|x_1, x_2, ...x_k) = \frac{1}{m} \sum_{j=1}^{m} P_{h_j}(c_i|x_1, x_2, ...x_k) \tag{2}$$

$$P_{Ensemble}(c_i|x_1, x_2, ...x_k) = \max_{j=1}^{m} P_{h_j}(c_i|x_1, x_2, ...x_k) \tag{3}$$

$$P_{Ensemble}(c_i|x_1, x_2, ...x_k) = \min_{j=1}^{m} P_{h_j}(c_i|x_1, x_2, ...x_k) \tag{4}$$

$$P_{Ensemble}(c_i|x_1, x_2, ...x_k) = med_{j=1}^{m} P_{h_j}(c_i|x_1, x_2, ...x_k) \tag{5}$$

$$P_{Ensemble}(c_i|x_1, x_2, ...x_k) = \Pi_{j=1}^{m} P_{h_j}(c_i|x_1, x_2, ...x_k) \tag{6}$$

All of the above rules can be generally referred to as algebraic fusion. In this context, the final classification through algebraic fusion is made by selecting the class that obtains the maximum posterior probability as defined in Eq. (7).

$$\begin{aligned} & assign \qquad c_t \rightarrow \{x_1, x_2, ..., x_k\}, \qquad if \\ & P_{Ensemble}(c_t|x_1, x_2, ...x_k) = \max_{i=1}^{n} P_{Ensemble}(c_i|x_1, x_2, ...x_k) \end{aligned} \tag{7}$$

In practice, the above rules of fusion, such as vote, mean, max and median, can be either fixed or trained. For example, majority vote is simply a fixed rule of fusion, whereas weighted vote is considered as a trained rule, since the weight for each base classifier needs to be estimated typically on validation data. Similarly, the algebraic rules can also be used by assigning a weight to each classifier. In addition, there are some other trained rules such as behaviour knowledge space (Huang and Suen, 1995), Dempster-Shafer (Xu et al., 1992), decision templates (Kuncheva et al., 2001).

From a theoretical perspective, trained rules are preferred to fixed rules leading to potentially better performance, since the classification performance achieved through using fixed rules are considered to be sub-optimal without measure of classifier weight, as discussed in Duin (2002). However, trained rules heavily need large and nicely cleaned data (Jr., 2011). In other words, trained rules are likely to result in overfitting if the data is small or not nicely cleaned.

In the context of fixed rules of fusion, some comparison studies have been done in Kittler et al. (1998); Kittler and Alkoot (2003); Kuncheva (2004); Tax et al. (2000, 1997). In general, these rules are considered to have overall good performance and high popularity in real applications (Kuncheva et al., 2001), and the majority vote and sum/mean rules are used the most frequently (Kittler and Alkoot, 2003). Also, the sum/mean rule is viewed as the most favourite one (Kittler et al., 1998; Kuncheva et al., 2001). These rules have been popularly used in pattern recognition (Mangai et al., 2010; Kamble and Kokate, 2017), e.g. handwriting digits recognition (Shukla and Pandey, 2014), characters recognition (Chackoa and P.M.Dhanya, 2015) and affect recognition (Gunes and Piccardi, 2005).

Overall, the above review of related works on ensemble learning (alongside the summary of methods analysis shown in Table 1) indicates that the existing ensemble learning approaches are generally designed to employ a single learning algorithm to train multiple classifiers on different data samples or feature subsets or to employ different learning algorithms to train multiple classifiers on the same feature set extracted from a single data sample, while only a single rule of fusion is adopted for making a final classification. In

Table 1: Theoretical analysis of different methods

Method	Pros	Cons
Bagging	variance reduction creation of diversity through different training samples	unsuitable for dealing with imbalanced data lack of diversity creation through different algorithms
Boosting	bias reduction creation of diversity through focusing on incorrectly classified instances at later stages of training	unsuitable for dealing with small data lack of diversity creation through different algorithms
Random subspace	reduction of curse of dimensionality creation of diversity through different feature subsets	no clear guideline on how to define the dimensionality of the feature subspace lack of diversity creation through different algorithms
Fixed rules of fusion	low computational complexity suitable for combining independent or lowly correlated classifiers with similar performance on small data	not suitable for combining highly correlated classifiers with different performance no consideration of the confidence of each single classifier
Trained rules of fusion	more suitable than fixed rules for combining highly correlated classifiers with different performance	high computational complexity unsuitable for dealing with small data

this way, the creation of diversity among classifiers can be limited, due to the case that the same algorithm may have different suitability for learning from different data samples or feature sets and that the same data sample or feature set may also show different suitability for different algorithms to learn effectively. In addition, the effectiveness of each fusion rule highly depends on the size of data and the actual degree of diversity among different classifiers. Therefore, it is necessary to design a more systematic framework of ensemble learning that produces an ensemble of ensembles and to incorporate different fusion rules into the framework, through employing the concepts of granular computing that will be introduced shortly in Section 2.2.

2.2. Overview of Granular Computing

Granular computing is a paradigm of information processing (Pedrycz and Chen, 2011, 2015a,b), which is aimed at structured thinking at the philosophical level and structured problem solving at the practical level (Yao, 2005b). Two main operations of granular computing are referred to as granulation and organization. The former operation is aimed to decompose a whole into parts, i.e. a top-down approach of information processing, whereas the latter

operation is aimed to integrate parts into a whole, i.e. a bottom-up approach of information processing (Liu et al., 2018; Yao, 2005a).

In granulation and organization, the main aim is to deal with information granules. Each granule is defined as a collection of smaller particles that can form a larger particle (Liu and Cocea, 2017b). In other words, different granules usually have different sizes, e.g. a set can be viewed as a granule, which can be either a smaller one or a larger one. Also, a smaller set can be an element (subset) of a larger set, which indicates the need to involve the concept of granularity, i.e. a smaller granule needs to be located at a lower level of granularity, whereas a larger granule should be located at a higher level of granularity (Liu and Cocea, 2018). In this context, granulation is essentially an operation of decomposing a larger granule at a higher level of granularity into several smaller granules at a lower level of granularity, and organization is essentially an operation in the opposite way.

In practice, granular computing concepts can be used in various application areas. For example, in natural language processing, a document is generally a complex instance that can be simplified through text parsing, i.e. the instance is defined as an information granule, which can be decomposed into sub-granules located at different levels of granularity, e.g. chapters, sections, paragraphs, sentences and words (Liu and Cocea, 2017b, 2018). Also, in image processing, a complex image usually needs segmentation into several target regions (Liu et al., 2018), which is viewed as a special form of information granulation. Other applications of granular computing concepts are popularly involved in multi-attribute decision making (Xu and Wang, 2016; Liu and You, 2017; Chatterjee and Kar, 2017; Zulueta-Veliz and Garca-Cabrera, 2018) and fuzzy sets (Chen and Chang, 2001; Chen, 1996), rough sets (Zhang et al., 2016b; Ma and Zhu, 2015) and clustering (Horng et al., 2005; Chen et al., 2009).

In the context of ensemble learning, an ensemble of classifiers can obviously be viewed as a granule. Also, as introduced in Liu and Cocea (2019), an ensemble can be larger to contain not only single classifiers but also smaller ensembles of classifiers, which shows that an ensemble can be created in a granular architecture involving different levels of granularity. On the other hand, the design of the Bagging approach shows the characteristics of granular computing. For example, this approach involves transforming a training set into different versions of samples, which is essentially a form of granulation (Liu and Cocea, 2017a). In the classification stage, all the classifiers in an ensemble need to be fused to provide a final classification for each in-

stance, which is essentially a form of organization (Liu and Cocea, 2017a). A very similar argumentation has also been made in Hu and Shi (2009).

3. Proposed Ensemble Learning Framework

In this section, we illustrate our proposed framework of ensemble learning and justify theoretically why the characteristics of the proposed framework can lead to advances in classification performance.

3.1. Key Features

The proposed framework essentially involves three parts of design, namely, data transformation, creation of primary ensembles and creation of secondary ensembles. The whole framework is illustrated in Fig. 2.

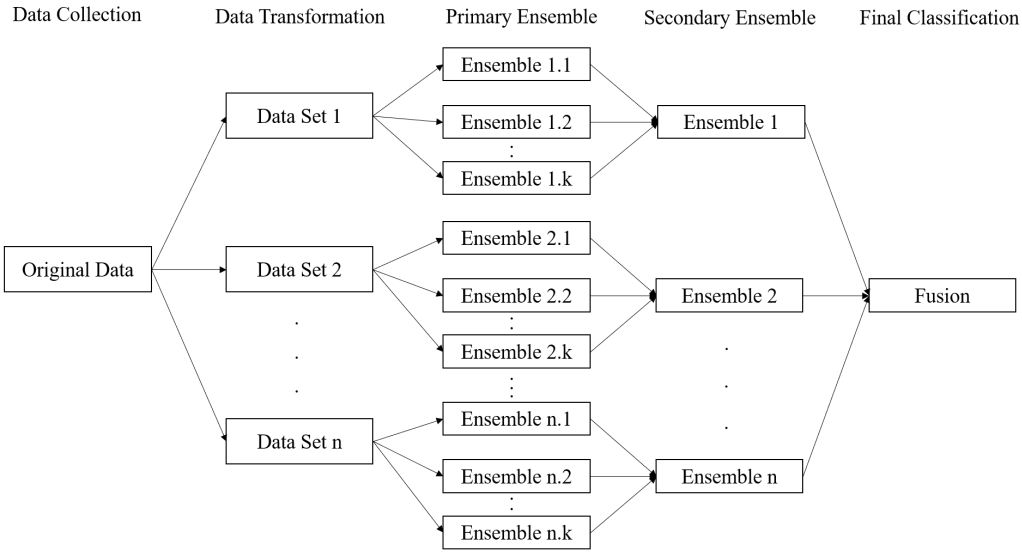

Figure 2: Proposed Framework of Ensemble Learning

In terms of data transformation, the aim is to transform a multi-class classification task into n binary classification tasks, where n is the number of classes. In this context, the original data set D needs to be transformed into n data sets $D[n]$ with manipulation on class labels $c[n]$. For example, if a data set contains three classes, namely, A, B and C, then the three manipulated data sets would each contain one of the above three classes and its negation, i.e. A and $\neg A$, B and $\neg B$, C and $\neg C$.

In the feature extraction stage, the manipulated data sets $D[n]$ would be transformed into the exactly same feature sets by using the same feature extraction method, since it is an unsupervised task. However, in order to obtain more diverse features, it is necessary to adopt multiple feature extraction methods, leading to different feature sets $F[m]$ extracted from each manipulated data set D_t for class c_t. The extracted feature sets $F[m]$ would usually need to be processed further for the purpose of feature selection, leading to removal of redundant features.

In the primary ensembles creation stage, a base classifier h_{tij} is trained on each feature set F_j extracted from data set D_t, by using algorithm a_i. For each data set D_t, the classifiers $h_{ti}[m]$ trained on different feature sets $F[m]$ by using the same algorithm a_i make up a primary ensemble E_{ti} through algebraic fusion. The whole procedure of primary ensembles creation is illustrated in Fig. 3.

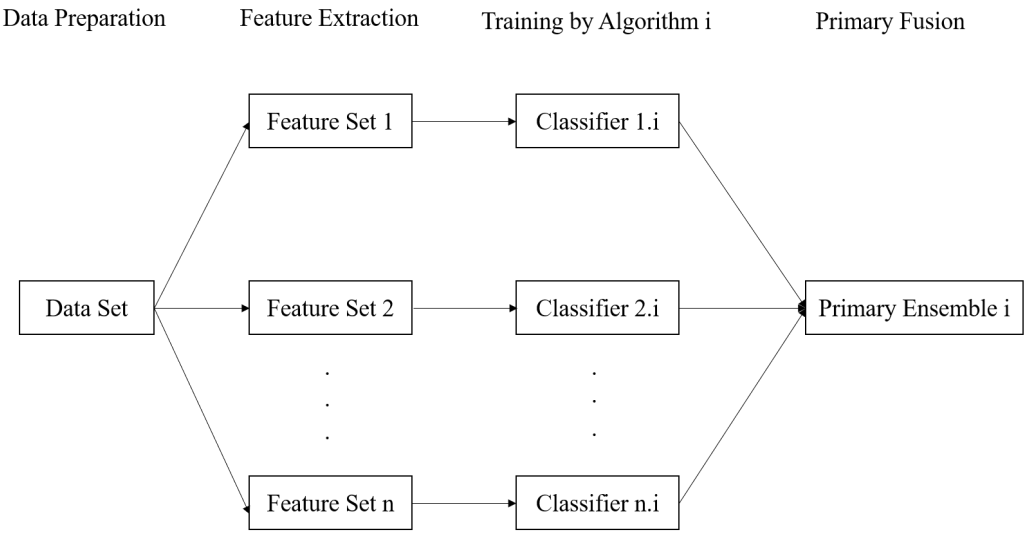

Figure 3: Procedure of Primary Ensembles Creation

Since each learning algorithm a_i can lead to creation of a primary ensemble E_{ti} resulting from manipulated data set D_t, it is very straightforward to create a secondary ensemble E_t through algebraic fusion of all the primary ensembles $E_t[k]$, which are created by using all the employed algorithms $a[k]$. In this context, n secondary ensembles $E[n]$ are obtained respectively for the n manipulated data sets $D[n]$.

As illustrated in Fig. 2, the final classification is made through fusion of all the secondary ensembles $E[n]$. The aim of the final fusion is to choose one of the predefined classes $c[n]$ to be the final output. Since each secondary ensemble consists of k primary ensembles of binary classifiers, it is necessary to check whether each secondary ensemble outputs a positive class c_t or a negative class $\neg c_t$. Ideally, it would be expected that only one secondary ensemble outputs a positive class, such that this class would be the final output. However, it is possible in reality that more than one ensemble output positive classes or even none of the ensembles output a positive class. In this case, it is necessary to check which ensemble has the highest confidence (posterior probability) for classifying the instance to the positive class, i.e. the positive class resulting from the most confident ensemble is chosen as the final output.

3.2. Justification

The proposed framework of ensemble learning is essentially designed in the setting of granular computing, which involves information granulation through transforming a multi-class classification task into n binary classification tasks. In other words, a multi-class classifier h, which is trained on the original data set D, is viewed as a granule g at a higher level of granularity. In this context, each binary classifier h_t, which is trained on a manipulated data set D_t, would be viewed as a sub-granule g_t (of granule g) at a lower level of granularity.

On the other hand, ensembles creation also involves the use of granular computing concepts. In particular, each primary ensemble E_{ti} is viewed as a granule at a lower level of granularity. The primary ensembles $E_t[k]$ created for a manipulated data set D_t make up a secondary ensemble E_t, which is viewed as a granule at a higher level of granularity. The creation of a secondary ensemble through algebraic fusion of primary ensembles is viewed as a special form of organization. Also, the final classification for an instance through fusion of all the secondary ensembles $E[n]$ is again viewed as a special form of organization. On the basis of the above justification, the ensemble learning framework is designed to achieve multi-level fusion (MLF), i.e. fusion operations at different levels of granularity.

In terms of data transformation, the main aim is to achieve training of classifiers in more depth. In particular, in a multi-class classification task, feature selection is aimed at identifying features that are capable of discriminating between classes. Also, classification is undertaken by training a clas-

sifier that discriminates one class from the other classes. However, when the number of classes is increased, it usually becomes increasingly more difficult to discriminate between classes. From this point of view, data transformation in the setting of granular computing leads to re-framing of the classification problem through training a binary classifier in more depth on each manipulated data set. In this context, feature selection can be done separately for each class in depth, i.e. a subset of relevant features is obtained for each class. Furthermore, a binary classifier is trained in depth on each feature subset selected for a specific class, and the binary classifiers are likely to be diverse, due to the case that they are trained on different feature subsets.

The above setting of data transformation can help reduce the risk of overfitting on one hand, through more effective feature selection for each class, since learning from features irrelevant for a specific class could affect the classification performance for this class. On the other hand, the transformation of a multi-class classification task into n binary classification tasks can result in the class imbalance issue, leading to the risk to affect the performance of feature selection and classification for a class of a very low frequency. However, this can be overcome in practice by selecting algorithms that are less sensitive to class imbalance for classifiers training. Also, our setting in the final stage (shown in Fig. 2) for transforming n binary classification tasks back into a multi-class classification task can also help avoid incorrect classification occurring from a binary classifier h_t due to the imbalance of the manipulated data set D_t, by fusing the outputs of the n binary classifiers to obtain a final classification for each instance.

As introduced in Section 3.1, classifiers training on the feature sets $F[m]$ extracted from each manipulated data set D_t can lead to k primary ensembles $E_t[k]$ and one secondary ensemble E_t. Since the base classifiers in each primary ensemble are trained on different feature sets by using the same algorithm, it is likely that the base classifiers are diverse. In other words, different feature sets are extracted by using different feature extraction methods, so the feature sets are likely to be diverse leading to more diverse classifiers being trained. Also, the primary ensembles are created by using different learning algorithms, which indicates that the primary ensembles are likely to be diverse, due to different strategies involved in these learning algorithms. However, the diversity creation in primary and secondary ensembles cannot guarantee an improvement of the classification performance through fusion of classifiers, especially when one classifier in an ensemble is dominant of the others. This indicates that ensemble creation needs to avoid having a

dominant classifier in the ensemble.

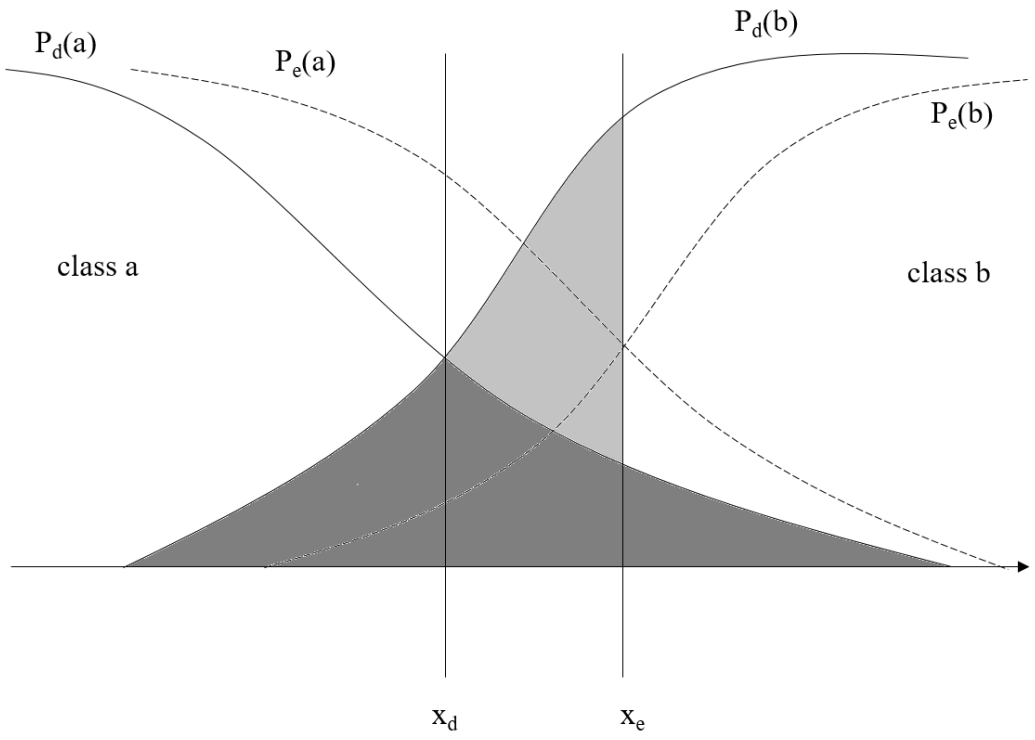

Figure 4: Tumer and Ghosh's framework (Brown and Kuncheva, 2010; Tumer and Ghosh, 1996a,b) for analyzing estimation errors

The motivation of using algebraic rules of classifiers fusion is essentially to reduce the estimation error of the posterior probability for each class towards reducing the risk of overfitting. In particular, as shown in Fig. 4, in a binary classification problem, the true probability distributions ($P_d(a)$ and $P_d(b)$) for classes a and b have an overlap (the dark shaded area in Fig. 4), which represents the Bayes error and is irreducible. However, due to the case that a training set is unlikely to represent a full population of a problem domain, the estimated posterior probabilities for the two classes ($P_e(a)$ and $P_e(b)$) are usually different from the true probabilities ($P_d(a)$ and $P_d(b)$), which leads to the added error (the light shaded area in Fig. 4).

In reality, it is fairly difficult to obtain a full population, so we adopt algebraic fusion rules for the purpose of reducing the estimation error of the posterior probability for each class, especially when the training sample

is small. Also, we choose to adopt fixed rules instead of trained rules for algebraic fusion, since the latter type of rules usually need a large amount of data for avoiding the case of overfitting, as mentioned in Section 2.1.

4. Experimental Studies, Results and Discussion

In this section, we report two experimental studies using a UCI data set on Sonar (Lichman, 2013) and an image data set on facial expression recognition, respectively, in order to evaluate our proposed framework of ensemble learning.

The first study is aimed at investigating the impacts of feature selection and multi-level fusion of classifiers on the classification performance. The 'Sonar' data set contains 60 features and 208 instances, where 97 instances belong to the 'Rock' class and 111 instances belong to 'Mine' class.

In the experimental setup, a feature subset, which contains 19 features, is created by using the Correlation-based Feature Subset Selection (CFS) method (Hall and Smith, 1997). In this context, each learning algorithm is used to train two classifiers respectively on the original feature set and the feature subset and the two classifiers make up a primary ensemble. In particular, we employ three popular learning algorithms, namely, Support Vector Machine (SVM), Multi-layer Perceptron (MLP) and K Nearest Neighbour (KNN), for training base classifiers. Therefore, there will be three primary ensembles created and each ensemble consists of two base classifiers trained by using one of the three learning algorithms. The base classifiers in each primary ensemble are fused by using the mean rule. Furthermore, the three primary ensembles are fused further by using the max rule, in order to make up the secondary ensemble for final classification of each new instance.

In terms of parameters setting of learning algorithms, trials and errors have been conducted for the parameter selections of the base classifiers. Specifically, SVM classifiers are trained with the overlapping penalty of 1.0 by using the polynomial kernel, and the values of bias, power and gama are all set to 1.0. The MLP architecture is set to have 1 hidden layer and 10 neurons per layer and classifiers are trained through up to 100 iterations. In terms of KNN, the value of K is set to 3 and the nearest neighbours are weighted according to their distances to the test instance. All the experiments are conducted using 10-fold cross validation.

The results are shown in Table 2. In this table, SVM1, MLP1 and KNN1 indicate that the three algorithms are used to train classifiers on the original

Table 2: Results on 'Sonar' Data Set

Method	Accuracy	F-measure (Rock)	F-measure (Mine)
SVM1	0.779	0.736	0.810
SVM2	0.774	0.759	0.787
SVM3	0.803	0.776	0.824
MLP1	0.808	0.792	0.821
MLP2	0.769	0.742	0.791
MLP3	0.808	0.785	0.826
KNN1	0.813	0.782	0.835
KNN2	0.856	0.840	0.868
KNN3	0.856	0.839	0.870
MLF	**0.870**	**0.852**	**0.884**

feature set, whereas SVM2, MLP2 and KNN2 indicate that the classifiers are trained on the feature subset created through using the CFS method. Furthermore, SVM3, MLP3 and KNN3 are created by fusing respectively the three pairs of classifiers: SVM1+SVM2, MLP1+MLP2 and KNN1+KNN2. Finally, MLF is created by fusing SVM3, MLP3 and KNN3.

The results shown in Table 2 indicate that feature selection through the CFS method leads to decrease of the classification performance for SVM and MLP but the performance is improved for KNN. In terms of primary ensembles creation, the fusion of SVM1 and SVM2 leads to a small improvement for both the accuracy and the F-measure for each class. However, for both MLP and KNN, the classifiers fusion leads to the same accuracy, marginal decrease of the F-measure for the 'Rock' class and a minor improvement for the the F-measure for the 'Mine' class, in comparison with the better one of the two base classifiers. In terms of the secondary ensemble creation, the fusion of the three primary ensembles leads to the best performance for both the accuracy and the F-measure for each class.

The above results show that the impact of feature selection on classifiers training is varied for different learning algorithms, i.e. for some algorithms, the overall performance of the classifiers trained on a feature subset may be increased, but the performance may be decreased for other algorithms. However, the increase or decrease of the overall performance can not simply indicate the increase or decrease of the classifier confidence for classifying each instance. From this point of view, it is likely to encourage the diversity

between the classifiers trained respectively on the original feature set and the feature subset, as supported by the results on diversity measures shown in Table 3.

Table 3: Diversity Measures for Classifiers in Primary Ensembles

Classifiers Pair	Q-statistics
SVM1+SVM2	0.918
MLP1+MLP2	0.711
KNN1+KNN2	0.785

The pairwise measures of diversity are achieved by using the Q-statistics, which is commonly used in practice (Kuncheva and Whitaker, 2003) and is defined in Eq. (8), i.e. the smaller the value the higher the diversity.

$$Q_{ij} = \frac{ad - bc}{ad + bc} \tag{8}$$

where a and d represent the numbers of instances that both classifiers h_i and h_j give the positive and negative prediction respectively; b represents the number of instances that classifier h_i gives the negative class but classifier h_j gives the positive class; and c represents the number of instances in the opposite case.

The results shown in Table 3 indicates that MLP and KNN lead to two pairs of classifiers of higher diversity, in comparison with SVM. However, the fusion of SVM1 and SVM2 leads to a larger improvement of the classification performance, which indicates that diversity is not a direct measure of the performance improvement as argued in Brown and Kuncheva (2010), but the encouragement of diversity between classifiers shows the effectiveness of advancing the classification performance for at least one class.

In the secondary ensemble creation stage, the fusion of the three primary ensembles leads to further encouragement as shown in Table 4.

Table 4: Diversity Measures for Primary Ensembles

Ensemble	SVM3	MLP3	KNN3
SVM3	1	0.825	0.706
MLP3	0.825	1	0.73
KNN3	0.706	0.73	1

The results shown in Table 4 indicate that the use of learning algorithms with different strategies can encourage more effectively the diversity between classifiers, leading to further advances in the classification performance.

The second study is aimed to investigate the impact of the transformation of a multi-class classification task into n binary classification tasks alongside multi-level fusion of binary classifiers. There are 593 sequences of frontal-view posed facial expression images contributed by 123 subjects in the CK+ data set (Lucey et al., 2010). We select a total of 344 (instances) peak facial expression images for the 7 (classes) expressions for the evaluation of the proposed framework. The frequency distribution among the 7 classes is 45 (angry): 59 (disgust): 25 (fear): 69 (happy): 35(neutral): 28 (sad): 83 (surprise).

In terms of feature extraction, the set of images is transformed into two sets of features, namely, Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP), where the former contains 575 features and the latter contains 530 attributes. Each of the extracted feature sets is assigned 7 different pairs of class labels (angry/not angry, disgust/not disgust, fear/not fear, happy/not happy, neutral/not neural, sad/not sad and surprise/not surprise) in order to create 7 new data sets for binary classification. In this context, each of the 7 classes (angry, disgust, fear, happy, neutral, sad and surprise) is treated as a target class, and the other 7 classes (not angry, not disgust, not fear, not happy, not neural, not sad and not surprise) are all non-target classes. Therefore, each of the 7 new data sets is created for identifying one of the 7 target classes.

Furthermore, each of the 14 data sets (7 with HOG features and 7 with LBP features) is processed through feature selection by using the CFS method, in order to remove redundant features and select a subset of features relevant for the target class only. In particular, the number of features selected for each data set is shown in Table 5.

In terms of training the base classifiers that make up primary ensembles, SVM, MLP and KNN are used as the learning algorithms with the same parameters setting as the one in Study 1 on the 'Sonar' data set. Again, all the experiments are conducted using 10-fold cross validation.

In terms of primary ensembles creation, a base classifier is trained on each of the 14 data sets, i.e. 14 base classifiers are trained in total, resulting from 7 HOG feature sets and 7 LBP feature sets, respectively. In this context, each learning algorithm would lead to 7 primary ensembles for the 7 target classes, respectively, and each ensemble consists of two classifiers trained,

Table 5: Number of features selected for each target class

Class	HOG	LBP
angry	40	14
disgust	47	25
fear	25	13
happy	56	29
neural	28	26
sad	17	10
surprise	53	39

respectively, on HOG and LBP features. Each primary ensemble is created through classifiers fusion in the mean rule. Furthermore, for each of the 7 target classes, a secondary ensemble is created by fusing in the median rule the three primary ensembles that result from the learning algorithms (SVM, MLP and KNN). Finally, all the secondary ensembles are fused to make a final classification for each instance.

In order to evaluate the performance of the proposed ensemble learning framework, we compare it with the standard learning methods (SVM, MLP and KNN) as well as the ensemble learning methods (RF and GBT), since they are all very popular for pattern recognition tasks. All the above methods are used to train classifiers on the HOG and LBP features in the setting of 7-class classification. Again, both the HOG and LBP feature sets are processed through feature selection by using the CFS method, leading to 71 HOG features and 39 LBP features being selected, respectively. The results are shown in Table 6.

Table 6: Results on Facial Emotions Recognition

Method	Accuracy	F1(angry)	F1(disgust)	F1(fear)	F1(happy)	F1(neutral)	F1(sad)	F1(surprise)
SVM1	0.814	0.644	0.891	**0.731**	**0.957**	0.545	**0.552**	0.952
SVM2	0.741	0.390	**0.941**	0.545	0.917	0.559	0.302	0.876
MLP1	0.762	0.583	0.885	0.582	0.942	0.516	0.441	0.897
MLP2	0.648	0.341	0.807	0.304	0.847	0.519	0.203	0.833
KNN1	0.747	0.454	0.870	0.5	0.932	0.507	0.348	0.947
KNN2	0.555	0.270	0.718	0.260	0.871	0.189	0.04	0.721
RF1	0.817	0.584	0.885	0.65	0.945	**0.685**	0.52	**0.964**
RF2	0.645	0.359	0.862	0.256	0.870	0.319	0.140	0.762
GBT1	0.765	0.522	0.840	0.553	0.889	0.638	0.528	0.927
GBT2	0.631	0.447	0.764	0.238	0.841	0.474	0.122	0.8
MLF	**0.834**	**0.703**	0.940	0.615	0.952	0.649	0.542	0.947

In Table 6, SVM1, MLP1, KNN1, RF1 and GBT1 indicate that these methods are used to train classifiers on the HOG feature set, whereas SVM2, MLP2, KNN2, RF2 and GBT2 indicate that classifiers training are done using the LBP feature set.

The results shown in Table 6 indicate that our proposed ensemble learning framework leads to the best overall accuracy of classification and F-measure for the 'angry' class. For all the other classes, the performance on F-measure is slightly worse than the best performing one, except for the 'fear' class. The low performance on the 'fear' class is likely due to the case that the lowest frequency of this class, leading to low performance of all the base classifiers apart from SVM1. As mentioned in Section 2.1, one of the two key points for effective ensemble learning is to make sure that each single classifier must not be too bad. In the above case, all the three learning algorithms (SVM, MLP and KNN) are not capable of training high quality classifiers on the LBP feature set and two of them can not train high quality classifiers on the HOG feature set, which indicates that it is unlikely to achieve advances in the performance through fusion of classifiers trained by using these algorithms.

The fearful facial expression images tend to indicate comparatively mild physical cues and facial deformations which also pose great challenges to other facial expression research (Zhang et al., 2016a, 2013). Nevertheless, for the overall accuracy for the 7-class expression recognition, the proposed meta-ensemble model achieves the highest performance and outperforms other ensemble models by a significant margin.

On the other hand, the results show that the best performing approach for each class is varied, e.g. the SVM classifier trained on the HOG feature set performs the best for the 'fear', 'happy' and 'sad' classes, whereas the RF ensemble trained on the HOG feature set is the best performing one for the 'neutral' and 'surprise' classes. Furthermore, although the classifiers trained on the LBP feature set perform worse than the ones trained on the HOG feature set, the results on the performance for the 'disgust' class show that the use of LBP features leads to a better SVM classifier being trained, in comparison with the use of HOG features. The above argumentation again indicates the necessity of adopting our proposed framework of ensemble learning, since the results show the effectiveness of reducing the variance of performance of the same classifier for different classes.

Furthermore, since convolutional neural networks (CNN) have been popularly used as the state of the art approaches in image recognition, we also compare our proposed ensemble learning approach with six pre-trained CNN

models, namely, GoogLeNet, Inceptionv3, ResNet101, AlexNet, VGG16 and VGG19, while the pre-trained CNN models are used in the setting of transfer learning. In particular, following the popular way of experimental evaluation as adopted in related works on image recognition through deep learning (Fernandes et al., 2018; Fielding and Zhang, 2018; Sun et al., 2018; Tan et al., 2019), the experiments are conducted using holdout validation over 10 runs by randomly selecting 90% of the instances for training and the rest for testing in each run. These deep networks are pre-trained using a million images and are able to classify images into 1000 object categories. We conduct the transfer learning by re-training the last learnable layer and the final classification layer in these deep networks using the CK+ dataset.

The setting of our proposed ensemble learning approach (alongside each standard learning algorithm used for training base classifiers) is the same as the one taken in the above 10-fold cross validation for obtaining the results shown in 6. The transfer learning based on each of the pre-trained CNN models is set as follows, i.e. MiniBatchSize (Size of the mini-batch)=10, the maximum number of Epochs=6, and learning rate=3e-4 with the stochastic gradient descent with momentum (SGDM) optimizer as the solver for training network.

Table 7: Classification Results through Holdout Validation

Method	Accuracy
GoogLeNet	0.788
Inceptionv3	0.636
ResNet101	0.758
AlexNet	0.789
VGG19	0.849
VGG16	0.849
MLF	**0.879**

The results obtained using holdout validation are shown in Table 7, which indicate that our proposed ensemble learning approach outperforms all the pre-trained CNN models adopted in the setting of transfer learning. The results indicate that our proposed approach shows better suitability for dealing with small data in comparison with deep learning approaches.

Overall, the results shown in Tables 2 and 6 indicate that the ways we designed the proposed ensemble learning framework can encourage more ef-

fectively the creation of diversity among different classifiers leading to better performance of classification, in comparison with popular ensemble learning approaches. Also, the proposed approach can achieve more stable performance for each single class, while the performance of each standard learning approach is likely to be good only for some but not all of the classes, through looking at the results on F-measure for each class as shown in Table 6. Moreover, the results shown in Table 7 indicate that it is necessary to explore in more depth the use of traditional learning approaches in the setting of ensemble learning, while the size of data is fairly small, due to the case that deep learning approaches highly need much larger data for achieving good learning performance.

5. Conclusion

In this paper, we proposed a systematic framework of ensemble learning in the setting of granular computing. In particular, this framework involves transforming a multi-class classification task into a number of binary classification tasks (through information granulation), which are finally turned back into a multi-class classification task to decide the final classification for each instance (through information organization). In the ensembles creation, the framework was designed to make the learning of binary classifiers benefit from diverse feature sets and learning algorithms, i.e. the ensemble learning task involves primary fusion of multiple classifiers trained for each class on different feature sets by the same learning algorithm and secondary fusion of the previously fused classifiers resulting from primary fusion.

We conducted experiments by using a UCI data set on 'Sonar' and the CK+ data set on facial expression recognition. The experimental results show that the proposed framework leads to considerable advances in the classification performance, in comparison with popular ensemble learning approaches (RF and GBT) as well as the standard learning algorithms (SVM, MLP and KNN) that were used to train classifiers that make up an ensemble. The experimental results obtained on the CK+ data set also show that our proposed ensemble learning approach outperforms transferring learning approaches based on CNN models pre-trained on image data in other domains.

In future, we will investigate the use of fuzzy set theory (Zadeh, 1965) for developing fuzzy ensemble learning approaches (Nakai et al., 2003), since fuzzy approaches are generally capable of dealing with ambiguous cases, e.g. the 'sad' and 'fear' emotions, in the setting of fuzzy expert systems for pat-

tern recognition. Also, we will investigate in depth the use of various feature selection techniques (Liu et al., 2018) for obtaining diverse feature sets towards training diverse classifiers and advancing further the performance of ensemble learning, i.e. designing multiple expert systems with high diversity. Moreover, it is also worth to investigate how to achieve the above-mentioned extraction of more diverse features in the setting of deep learning frameworks. In addition, we will look to incorporate collaborative characteristics into the proposed framework of ensemble creation, such that the design of each of multiple expert systems can be enhanced through collaborations with the designs of the other expert systems.

Acknowledgements

The authors acknowledge support from the Social Data Science Lab at the Cardiff University and the Affective and Smart Computing Research Group at the Northumbria University. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

G. Brown and L. I. Kuncheva. Good and bad diversity in majority vote ensembles. In *International Workshop on Multiple Classifier Systems*, pages 124–133, Cairo, Egypt, 7-9 April 2010. Springer.

A. M. M. Chackoa and P.M.Dhanya. Multiple classifier system for offline malayalam character recognition. *Procedia Computer Science*, 46(4):86–92, 2015.

K. Chatterjee and S. Kar. Unified granular-number-based ahp-vikor multicriteria decision framework. *Granular Computing*, 2(3):199–221, 2017.

S.-M. Chen. A fuzzy reasoning approach for rule-based systems based on fuzzy logics. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 26(5):769–778, 1996.

S.-M. Chen and T.-H. Chang. Finding multiple possible critical paths using fuzzy pert. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 31(6):930–937, 2001.

S.-M. Chen, N.-Y. Wang, and J.-S. Pan. Forecasting enrollments using automatic clustering techniques and fuzzy logical relationships. *Expert Systems with Applications*, 36(8):11070–11076, 2009.

R. P. Duin. The combining classifier: to train or not to train? In *Object recognition supported by user interaction for service robots*, pages 765–770, Quebec, Canada, 11-15 August 2002. IEEE.

K. Fernandes, R. Cruz, and J. S. Cardoso. Deep image segmentation by quality inference. In *Proceedings of 2018 International Joint Conference on Neural Networks*, pages 1–8, Rio de Janeiro, Brazil, 8-13 July 2018.

B. Fielding and L. Zhang. Evolving image classification architectures with enhanced particle swarm optimisation. *IEEE Access*, 6:68560–68575, 2018.

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, Bari, Italy, 3-6 July 1996.

H. Gunes and M. Piccardi. Affect recognition from face and body: Early fusion vs. late fusion. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 3437–3443, Quebec, Canada, 12-12 October 2005. IEEE.

M. A. Hall and L. A. Smith. Feature subset selection: a correlation based filter approach. In *1997 International Conference on Neural Information Processing and Intelligent Information Systems*, pages 855–858, Berlin, Germany, 1997. Springer.

T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8): 832–844, 1998.

Y.-J. Horng, S.-M. Chen, Y.-C. Chang, and C.-H. Lee. A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE Transactions on Fuzzy Systems*, 13(2):216–228, 2005.

H. Hu and Z. Shi. Machine learning as granular computing. In *IEEE International Conference on Granular Computing*, pages 229–234, Nanchang, Beijing, 17-19 August 2009.

Y. S. Huang and C. Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):90–94, 1995.

M. P. P. Jr. Combining classifiers: from the creation of ensembles to the decision fusion. In *24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials*, pages 1–10, Alagoas, Brazil, 28-30 August 2011. IEEE.

V. V. Kamble and R. D. Kokate. Review on multiple classifier system in pattern recognition. *International Journal of Innovative Research in Science, Engineering and Technology*, 6(1):284–290, 2017.

J. Kittler and F. M. Alkoot. Sum versus vote fusion in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):281–285, 2003.

J. Kittler, M. Hatef, R. P. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3): 226–239, 1998.

I. Kononenko and M. Kukar. *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited, Chichester, 2007.

L. I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):281–285, 2002.

L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley and Sons, Inc., New Jersey, 2004.

L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

L. I. Kuncheva, J. C. Bezdek, and R. P. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34: 299–314, 2001.

J. Li and L. Wong. Rule-based data mining methods for classification problems in biomedical domains. In *A tutorial note for the 15th European Conference on Machine Learning (ECML) and the 8th European Conference on Principles and Practice for Knowledge Discovery in Databases (PKDD)*, pages 1–31, Pisa, Italy, 20-24 September 2004.

M. Lichman. UCI machine learning repository, http://archive.ics.uci.edu/ml, 2013.

H. Liu and M. Cocea. Granular computing based approach for classification towards reduction of bias in ensemble learning. *Granular Computing*, 2(3): 131–139, 2017a.

H. Liu and M. Cocea. Fuzzy information granulation towards interpretable sentiment analysis. *Granular Computing*, 2(4):289–302, 2017b.

H. Liu and M. Cocea. *Granular Computing Based Machine Learning: A Big Data Processing Approach*. Springer, Berlin, 2018.

H. Liu and M. Cocea. Nature inspired framework of ensemble learning in granular computing context. *Granular Computing*, In press, 2019.

H. Liu and A. Gegov. *Collaborative Decision Making by Ensemble Rule Based Classification Systems*, chapter 10, pages 245–264. Springer, Switzerland, 2015.

H. Liu, A. Gegov, and M. Cocea. *Rule Based Systems for Big Data: A Machine Learning Approach*. Springer, Switzerland, 2016.

H. Liu, M. Cocea, and W. Ding. Multi-task learning for intelligent data processing in granular computing context. *Granular Computing*, 3(3):257–273, 2018.

P. Liu and X. You. Probabilistic linguistic todim approach for multiple attribute decision-making. *Granular Computing*, 2(4):332–342, 2017.

P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete expression dataset for action unit and emotion-specified expression. In *Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis*, pages 94–101, San Francisco, USA, 13-18 June 2010.

D. Ma and W. Zhu. A matroidal structure for formal context and its applications on epidemiological study. In *International Conference on Machine Learning and Cybernetics*, pages 93–98, Guangzhou, China, 12-15 July 2015.

U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury. A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical Review*, 27(4):293–307, 2010.

P. Melville and R. J. Mooney. Creating diversity in ensembles using artificial data. *Information Fusion*, 6(1):99–111, 2005.

G. Nakai, T. Nakashima, and H. Ishibuchi. A fuzzy ensemble learning method for pattern classification. *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, 15(6):671–681, 2003.

J. O. Ogutu, H.-P. Piepho, and T. Schulz-Streeck. A comparison of random forests, boosting and support vector machines for genomic selection. *BMC Proceedings*, 5(3):S11, May 2011. ISSN 1753-6561. doi: 10.1186/1753-6561-5-S3-S11. URL `https://doi.org/10.1186/1753-6561-5-S3-S11`.

W. Pedrycz and S. M. Chen. *Granular Computing and Intelligent Systems: Design with Information Granules of Higher Order and Higher Type*. Springer, Heidelberg, 2011.

W. Pedrycz and S. M. Chen. *Information Granularity, Big Data, and Computational Intelligence*. Springer, Heidelberg, 2015a.

W. Pedrycz and S. M. Chen. *Granular Computing and Decision-Making: Interactive and Iterative Approaches*. Springer, Heidelberg, 2015b.

S. K. Shukla and A. Pandey. Classification of devnagari numerals using multiple classifier. *International Journal of Computer Trends and Technology*, 12(4):196–200, 2014.

Y. Sun, B. Xue, M. Zhang, and G. G. Yen. A particle swarm optimization-based flexible convolutional autoencoder for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2018.

P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 2005.

T. Y. Tan, L. Zhang, C. P. Lim, B. Fielding, Y. Yu, and E. Anderson. Evolving ensemble models for image segmentation using enhanced particle swarm optimization. *IEEE Access*, 7:34004–34019, 2019.

D. M. Tax, R. P. Duin, and M. van Breukelen. Comparison between product and mean classifier combination rules. In *In Proc. Workshop on Statistical Pattern Recognition*, pages 165–170, 1997.

D. M. Tax, M. van Breukelen, R. P. Duin, and J. Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition*, 33:1475–1485, 2000.

K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4):385–403, 1996a.

K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341–348, 1996b.

L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions ON Systems, Man, and Cybernetics*, 22(3):418–435, 1992.

Z. Xu and H. Wang. Managing multi-granularity linguistic information in qualitative group decision making: an overview. *Granular Computing*, 1 (1):21–35, 2016.

J. Yao. Information granulation and granular relationships. In *IEEE International Conference on Granular Computing*, pages 326–329, Beijing, China, 25-27 July 2005a.

Y. Yao. Perspectives of granular computing. In *Proceedings of 2005 IEEE International Conference on Granular Computing*, pages 85–90, Beijing, China, 25-27 July 2005b.

L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

L. Zhang, M. Jiang, D. Farid, and M. A. Hossain. Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot. *Expert Systems with Applications*, 40(13):5160–5168, 2013.

L. Zhang, K. Mistry, S. C. Neoh, and C. P. Lim. Intelligent facial emotion recognition using moth-firefly optimization. *Knowledge-Based Systems*, 111:248–267, 2016a.

Q. Zhang, Q. Xie, and G. Wang. A survey on rough set theory and its applications. *CAAI Transactions on Intelligence Technology*, 1(4):323–333, 2016b.

Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, London, 2012.

Y. Zulueta-Veliz and L. Garca-Cabrera. A choquet integral-based approach to multiattribute decision-making with correlated periods. *Granular Computing*, 3(3):245–256, 2018.